Thème : Représentation des données	Activité : Par dix, par deux, par seize…	NSI 1ère

« Il n'y a que 10 sortes de personnes : celles qui connaissent le binaire et les autres. »

La première préoccupation de l'Informatique est de recueillir des informations, qui peuvent prendre des formes très diverses et que l'on cherche à *représenter* afin de les traiter de manière automatisée.
Le type de plus simple d'informations que nous pouvons recueillir sont des nombres.

Partie I : Une petite histoire des nombres

Document 1 : Les petits cailloux

On a tous eu un jour l'occasion de compter une quantité importante de petits objets : des pièces de monnaie, des billes, des cartes… Et la façon la plus simple de compter est de le faire un par un.
On peut donc rapidement voir que, de manière primitive, on peut représenter les nombres entiers naturels en utilisant un symbole unique. Pour représenter un nombre entier n, il suffit alors d'écrire à la suite n fois le symbole choisi. Ce système de représentation est appelé système unaire (basé sur un symbole).

I.1. Représenter les nombres suivant en utilisant le symbole donné :
- **a.** Nombre sept avec des étoiles ☆
- **b.** Nombre trente-deux avec des cœurs ♥

I.2. La nature du symbole a-t-elle une importance sur la représentation ?

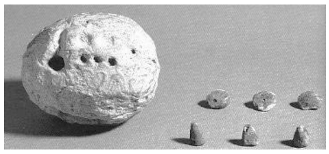
I.3. Quelle est la principale limitation de ce système ?

Une façon ingénieuse de simplifier les comptes est de compter par paquet. Le nombre d'unité dans chaque paquet correspond alors à la *base* de représentation.

Document 2 : De la Mésopotamie à l'Empire Maya

Mésopotamiens (-3500 avant J.C.)

L'un des premiers systèmes de numération était constitué de petits jetons de terre cuite de formes et de tailles différentes suivant la quantité qu'ils représentent. Ces jetons étaient emprisonnés dans une boule creuse en argile, ce qui permettait de vérifier que les transactions commerciales étaient exactes. Ces petits cailloux, dont le nom latin est *calculi*, sont à l'origine du mot calcul !

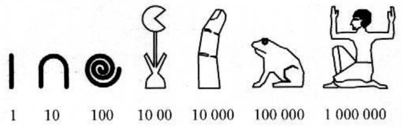


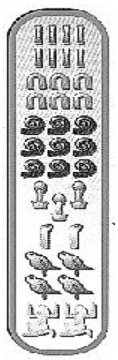
- Petit cône = 1
- Petite bille = 10
- Grand cône = 60
- Grand cône percé = 600
- Grosse bille = 3600
- Grosse bille percée = 36000

Ce système numérique compte par « paquet » de 60, ce qui peut paraitre incongru mais qui, en fait, se base sur le même principe que le compte en base 10 : compter sur ses mains. Il s'agit en fait d'une combinaison entre les 5 doigts de la main gauche et les phalanges des quatre doigts de la main droite (le pouce sert à compter les 12 phalanges). C'est ce système que l'on retrouve aujourd'hui au travers des unités de temps.

Egyptiens (IIIe siècle avant J.C.)

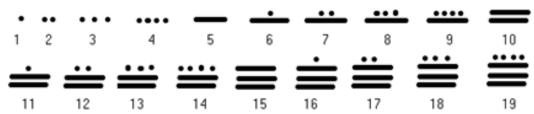
Les scribes égyptiens écrivent les nombres sur des papyrus sous forme de hiéroglyphes, les nombres se notant par répétition de signes, en base 10.





Maya (VIe siècle)

Les **mayas, qui sont les inventeurs du calendrier,** se servent des nombres pour calculer le temps. Leur système de numération suit le principe de position dans la base 20 et trouve ses origines avec nos 10 doigts et 10 orteils !



Source : http://www.maths-et-tiques.fr/index.php/histoire-des-maths/nombres/histoire-des-nombres

I.4. Quelle sont les différentes bases présentées dans le document 2 ? Sur quel principe se basent-elles ?

I.5. Quel nombre est représenté dans le cartouche égyptien ? Détailler le raisonnement.

Partie II : Une histoire de bases

A. Base décimale

Pour comprendre la représentation des nombres dans différentes bases, commençons par étudier la base décimale occidentale que nous connaissons bien.

IIA.1. Dans le nombre 1976, décrire à quoi correspond chaque symbole (« 1 », « 9 », « 7 » et « 6 ») par rapport à la position qu'il possède.

IIA.2. Quelle est alors, en puissances de 10, la décomposition de 1976 ? Et 76 341 ? Et 240 120 ?

IIA.3. Compléter la définition de la représentation des nombres en base décimale ci-dessous :

> Un entier naturel N se décompose en **base décimale (10)** grâce à une suite de symboles $\{n_p ; n_{p-1} ;\dots ; n_0\}$ choisis dans l'ensemble {0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9} avec $n_p \neq 0$, de telle sorte que :
>
> $$N = n_0 \times \dots\dots + n_1 \times \dots\dots + n_2 \times \dots\dots + \dots + n_{p-1} \times \dots\dots + n_p \times \dots\dots$$
>
> On écrit alors ce nombre $N = \overline{n_p n_{p-1} \dots n_2 n_1 n_0}^{10}$ et $10^{p+1} > N \geq 10^p$

B. Base binaire

Pour représenter un nombre en base 10, il faut donc disposer de dix symboles, ce qui peut être problématique lorsque c'est une machine qui compte (la machine n'a pas de doigts !). Mais cette dernière peut comprendre « allumé » et « éteint » par exemple. On représente alors les nombres en base 2, appelée base binaire. Par convention, on utilise les symboles 0 et 1.

IIB.1. Compléter la définition de la représentation binaire ci-dessous.

> Un entier naturel N se décompose en **base binaire (2)** grâce à une suite de symboles $\{n_p ; n_{p-1} ;\dots ; n_0\}$ choisis dans l'ensemble {.................................} avec $n_p \neq 0$, de telle sorte que :
>
> N = ...
>
> On écrit alors ce nombre .... et ....> N ≥ ....

IIB.2. Donner la représentation décimale des nombres binaires suivants.

a. $\overline{10}^2$ **b.** $\overline{11}^2$ **c.** $\overline{101}^2$ **d.** $\overline{10110}^2$

e. $\overline{1110}^2$ **f.** $\overline{100000010}^2$ **g.** $\overline{1111111}^2$

h. Il existe une manière plus rapide/efficace de déterminer la représentation décimale du nombre $\overline{1111111}^2$. Voyez-vous laquelle ?

Le binaire en informatique :

> On peut mesurer la taille de la représentation binaire d'un nombre en comptant le nombre de caractères utilisés. En anglais, « caractères binaires » se dit « *bi*nary digi*t* », on appelle donc chacun des caractères binaires ***bit***. La valeur que prend chaque bit est appelé ***poids du bit***.
>
> Les premiers ordinateurs manipulaient des « cases mémoires » de taille 8 bits. Cela a poussé les informaticiens à mesurer les fichiers en nombre de cases mémoires, soit en nombre de paquets de 8 bits : des ***octets***.

Maintenant que l'on sait passer de la base 2 à la base 10, on va chercher à faire l'inverse : passer de la base 10 à la base 2.

• **Première méthode**

Lorsque le nombre N à convertir en base décimal est « petit », on peut se référer directement aux différentes valeurs de la base binaire.

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	2	4	8	16	32	64	128	256

Le nombre de caractères nécessaire à la représentation binaire d'un nombre N correspond à la plus petite puissance p de 2 telle que $N < 2^p$.

On cherche alors la puissance de 2 directement inférieure au nombre N, on la soustrait de N et on recommence l'opération avec le reste de la soustraction.

Exemple : Le nombre décimal 155

Comme $155 < 2^8$, on sait que sa représentation binaire sera composée de 8 caractères.

$155 \geq 2^7$, donc on retient $1{\times}2^7$

Il reste $155 - 128 = 27$; $27 \geq 2^4$, donc on retient $1{\times}2^4$

Il reste $27 - 16 = 11$; $11 \geq 2^3$, donc on retient $1{\times}2^3$

Il reste $11 - 8 = 3$; $3 \geq 2^1$, donc on retient $1{\times}2^1$

Il reste $3 - 2 = 1$; $1 \geq 2^0$, donc on retient $1{\times}2^0$

La représentation binaire de 155 est donc $\overline{10011011}^2$

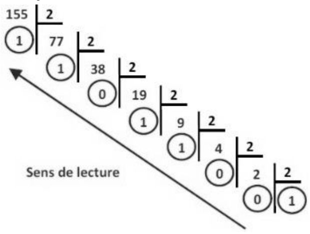
1	0	0	1	1	0	1	1
$\times 2^7$	$\times 2^6$	$\times 2^5$	$\times 2^4$	$\times 2^3$	$\times 2^2$	$\times 2^1$	$\times 2^0$

• **Deuxième méthode**

Pour passer de la représentation décimale à la représentation binaire, on procède à une succession de divisions euclidienne par 2, et ce jusqu'à ce que le quotient soit strictement inférieur à 2, en notant bien le reste de la division à chaque fois. Une fois le quotient final (<2) atteint, on peut lire directement la représentation du nombre en binaire, en partant du quotient final et en remontant de reste en reste.

Lorsque le nombre N est grand, il est plus efficace d'utiliser cette méthode.

Exemple : Le nombre décimal 155



Ainsi, $155 = \overline{10011011}^2$

IIB.3. Donner la représentation binaire des nombres décimaux suivants.

a. 9 **b.** 24 **c.** 94 **d.** 401

e. 257 **f.** 1026 **g.** 1023

C. Base hexadécimale

Un entier naturel N se décompose en **base hexadécimale (16)** grâce à une suite de symboles $\{n_p ; n_{p-1} ;\ldots ; n_0\}$ choisis dans l'ensemble {0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9 ; A ; B ; C ; D ; E ; F }*, avec $n_p \neq 0$, de telle sorte que :

$$N = n_0\times16^0 + n_1\times16^1 + n_2\times16^2 + \ldots + n_{p-1}\times16^{p-1} + n_p\times16^p$$

On écrit alors ce nombre $N=\overline{n_p n_{p-1}\ldots n_2 n_1 n_0}^{16}$ et $16^{p+1} > N \geq 16^p$

* où A ; B ; C ; D ; E ; F représentent respectivement les entiers 10 ; 11 ; 12 ; 13 ; 14 ; 15

IIC.1. Donner la représentation décimale des nombres hexadécimaux suivants.

a. $\overline{1B}^{16}$ **b.** $\overline{324}^{16}$ **c.** $\overline{A201E2}^{16}$

IIC.2. Donner la représentation hexadécimale des nombres décimaux suivants.

a. 8 **b.** 12 **c.** 120 336 **d.** 3 201 562

Ce qu'il faut retenir

❖ Bases décimale (10)

Un entier naturel N se décompose en base décimale grâce à une suite de symboles $\{n_p ; n_{p-1} ;\dots ; n_0\}$ choisis dans l'ensemble {0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9} avec $n_p \neq 0$, de telle sorte que :

$$N = n_0\times10^0 + n_1\times10^1 + n_2\times10^2 + \dots + n_{p-1}\times10^{p-1} + n_p\times10^p$$

On écrit alors ce nombre $N=\overline{n_pn_{p-1}\dots n_2n_1n_0}^{10}$ et $10^{p+1} > N \geq 10^p$

❖ Bases binaire (2)

Un entier naturel N se décompose en base binaire grâce à une suite de symboles $\{n_p ; n_{p-1} ;\dots ; n_0\}$ choisis dans l'ensemble {0 ; 1} avec $n_p \neq 0$, de telle sorte que :

$$N = n_0\times2^0 + n_1\times2^1 + n_2\times2^2 + \dots + n_{p-1}\times2^{p-1} + n_p\times2^p$$

On écrit alors ce nombre $N=\overline{n_pn_{p-1}\dots n_2n_1n_0}^{2}$ et $2^{p+1} > N \geq 2^p$

❖ Binaire en informatique

On peut mesurer la taille de la représentation binaire d'un nombre en comptant le nombre de caractères utilisés. En anglais, « caractères binaires » se dit « *bi*nary digi*t* », on appelle donc chacun des caractères binaires *bit*. La valeur que prend chaque bit est appelé *poids du bit*.

Les premiers ordinateurs manipulaient des « cases mémoires » de taille 8 bits. Cela a poussé les informaticiens à mesurer les fichiers en nombre de cases mémoires, soit en nombre de paquets de 8 bits : des *octets*.

❖ Bases hexadécimale (16)

Un entier naturel N se décompose en base hexadécimale grâce à une suite de symboles $\{n_p ; n_{p-1} ;\dots ; n_0\}$ choisis dans l'ensemble {0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9 ; A ; B ; C ; D ; E ; F }*, avec $n_p \neq 0$, de telle sorte que :

$$N = n_0\times16^0 + n_1\times16^1 + n_2\times16^2 + \dots + n_{p-1}\times16^{p-1} + n_p\times16^p$$

On écrit alors ce nombre $N=\overline{n_pn_{p-1}\dots n_2n_1n_0}^{16}$ et $16^{p+1} > N \geq 16^p$

* où A ; B ; C ; D ; E ; F représentent respectivement les entiers 10 ; 11 ; 12 ; 13 ; 14 ; 15

Savoir faire

❖ Conversion décimale-binaire

• Première méthode

Lorsque le nombre N à convertir en base décimal est « petit », on peut se référer directement aux différentes valeurs de la base binaire.

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	2	4	8	16	32	64	128	256

Le nombre de caractères nécessaire à la représentation binaire d'un nombre N correspond à la plus petite puissance p de 2 telle que $N < 2^p$.

On cherche alors la puissance de 2 directement inférieure au nombre N, on la soustrait de N et on recommence l'opération avec le reste de la soustraction.

• Deuxième méthode

Pour passer de la représentation décimale à la représentation binaire, on procède à une succession de divisions euclidienne par 2, et ce jusqu'à ce que le quotient soit strictement inférieur à 2, en notant bien le reste de la division à chaque fois. Une fois le quotient final (<2) atteint, on peut lire directement la représentation du nombre en binaire, en partant du quotient final et en remontant de reste en reste.

Lorsque le nombre N est grand, il est plus efficace d'utiliser cette méthode.